

# Covid 19

Abanoub George, Marc Ashraf, Alaa Yehia, Salma Mohamed, Omar Badry

June 2020

Department of Computer Science, Misr International University (MIU)

Email: abanoub1703340@miuegypt.edu.eg

Email: mark1712199@miuegypt.edu.eg

Email: alaa1709041@miuegypt.edu.eg

Email: salma1711570@miuegypt.edu.eg

Email: omar1707863@miuegypt.edu.eg

GitHub: <https://github.com/BonyGeorge/Artificial-Intelligence-Project->

# Contents

<b>1</b>	<b>Abstract</b>	<b>3</b>
1.1	Keywords . . . . .	3
<b>2</b>	<b>Introduction</b>	<b>3</b>
<b>3</b>	<b>Literature survey</b>	<b>4</b>
<b>4</b>	<b>Proposed COVID-19 system</b>	<b>5</b>
4.1	First Dataset . . . . .	5
4.1.1	KNN algorithm using first dataset . . . . .	6
4.1.2	Decision tree using First Dataset . . . . .	8
4.2	Second Dataset . . . . .	9
4.2.1	K-means algorithm using second dataset . . . . .	10
4.2.2	Support vector machines algorithm using second dataset .	10
<b>5</b>	<b>Result and Discussion</b>	<b>11</b>
5.1	KNN algorithm . . . . .	11
5.2	Decision tree algorithm . . . . .	12
5.3	K-means Algorithm . . . . .	13
5.4	SVM algorithm . . . . .	15
<b>6</b>	<b>Conclusion</b>	<b>17</b>
<b>7</b>	<b>Other References</b>	<b>18</b>

## List of Figures

1	First Dataset Part 1 . . . . .	5
2	First Dataset Part 2 . . . . .	6
3	First Dataset Part 3 . . . . .	6
4	KNN algorithm code Part 1 . . . . .	7
5	KNN algorithm code Part 2 . . . . .	8
6	KNN algorithm code Part 3 . . . . .	8
7	Second Dataset Part 1 . . . . .	9
8	Second Dataset Part 2 . . . . .	9
9	Second Dataset Part 3 . . . . .	10
10	KNN Results Part 1 . . . . .	11
11	KNN Results Part 2 . . . . .	11
12	Decision tree algorithm results with split 80% . . . . .	12
13	Decision tree algorithm results with split 50% . . . . .	12
14	K-means results Part 1 . . . . .	13
15	K-means results Part 2 . . . . .	13
16	K-means results Part 3 . . . . .	14
17	K-means results Part 4 . . . . .	14
18	K-means results visualize . . . . .	15
19	SVM results : Time taken to build model . . . . .	15
20	SVM results : Comparing between USA and Russia . . . . .	16
21	SVM results : Comparing between USA and Ivory Coast . . . . .	16

# 1 Abstract

COVID-19 was first established in Wuhan city in China, then it spreaded worldwide to alot of countries. The number of infected countries differ from one to another according to the medications and the number of population. To help us know the number of infected people of COVID-19 in each country, we used two different datasets and four different algorithms which are KNN, SVM, K means, Decision tree. .The first dataset used was about countries and the number predicting infected patients and the real number of infected patients of the current time,this dataset was used in KNN and Decision tree algorithms. KNN algorithm was used by Python. The second dataset used was about 210 countries with their continent, population, total cases, total deaths, total recovered, new death and new recovered. This dataset was used in SVM and K-means algorithms. SVM, K-means, Decision tree algorithms was used by Weka.

## 1.1 Keywords

COVID-19, KNN, K-means, SVM, Decision tree, Python, Weka.

# 2 Introduction

COVID-19 virus was first established in Wuhan city in China in December 2019 and it quickly spread worldwide. In March 2020, Most of the infected countries closed their airlines and factories and the education had stopped due to the huge spreadness of Corona virus. As of Feb 24, 2020, there was about 80,000 approved cases have been announced in more than 28 countries.[3][8][4][2] To this moment, No drugs or vaccine or supplement have been detected to get rid of the virus from the body.

The used algorithms in this project are KNN, SVM, K means, Decision tree.

1. KNN: It resembles K nearest neighbors in which it stores all obtainable cases then it distinguishes new cases based on similarity measures.It is responsible of recognition and regression.
2. SVM: SVM algorithm is used in classification and regression problems. In SVM algorithm, each data item is set as a point in n-dimensional space, with the value of each feature being the value of a particular coordinate.
3. K means: K means algorithm is considered as a unsupervised algorithm. It is also considered a method of vector quantization.K means work as follow, First it identifies the center of cluster,Then it assigns the nearest cluster to data point.Then setting the position of each cluster.

4. Decision tree: Decision tree algorithm is used for solving classification and regression problems. The purpose of using this algorithm is to create a trained dataset which will be used to know the value of the target.

Our goal in this document is very simple, which is to compare between the number of infected people in each country, and then study how every country held this situation. Although none of us have a medical background, We tried our best to study the situation of COVID-19 in each country and its numbers.

### 3 Literature survey

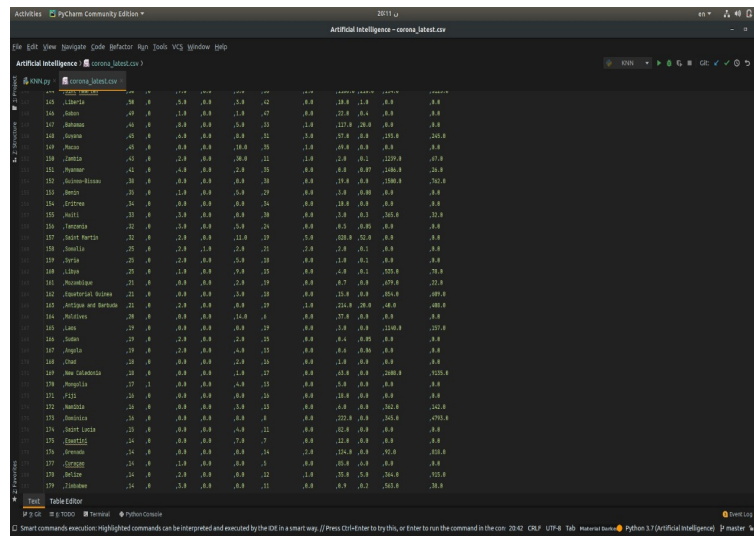
Alot of researchers made researches on this issue especially the issue is spread worldwide and it is still brand new. The first research [7] was made by Max Roser, Hannah Ritchie and Esteban Ortiz-Ospina. This research discusses the statistics of the spread of COVID-19 which takes its information from WHO(world health organization). The second research [5] was made by Mary A Lake in March 2020 and she talked about the current knowledge of COVID-19. The third research [1] was made by (Mohamad Chahrour, Sahar Assi, Michael Bejjani, Ali A Nasrallah, Hamza Salhab, Mohamad Fares, and Hussein H Khachfe ) which talked about COVID-19 since its discovery since December 2019. The last research [6] talked about the raise of number of corona virus worldwide and its symptoms.

## 4 Proposed COVID-19 system

We used two different datasets as we used many different algorithms.

### 4.1 First Dataset

First the dataset we used is a dataset of countries with its infected patients in each country. This dataset was made recently. This dataset was used for **KNN** algorithm. These datasets is based on when entering a test data, it compares the real data (test data) with the predicted ones which was already in the dataset.



The screenshot shows a PyCharm IDE window with a file named 'Artificial Intelligence - corona\_latest.csv'. The data is presented in a table with columns for country names and various numerical values. The table is sorted by the first column, which lists countries like Liberia, Albania, Romania, Kosovo, Pakistan, Serbia, Myanmar, Azerbaijan, Mexico, Nepal, Cyprus, Haiti, Tanzania, Saint Helena, Somalia, Jordan, China, Philippines, Kazakhstan, Antigua and Barbuda, Maldives, Laos, Japan, Angola, Chad, New Zealand, Mongolia, Fiji, Jamaica, Saint Lucia, Georgia, Armenia, Cyprus, Belize, and Zimbabwe.

Country	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7	Value 8	Value 9	Value 10
Liberia	58	0	1.0	0.0	1.0	10	0.0	10.0	1.0	0.0
Albania	40	0	1.0	0.0	1.0	10	0.0	10.0	0.0	0.0
Romania	40	0	0.0	0.0	1.0	10	1.0	10.0	0.0	0.0
Kosovo	40	0	0.0	0.0	0.0	10	1.0	10.0	0.0	10.0
Pakistan	40	0	0.0	0.0	10.0	10	1.0	10.0	0.0	0.0
Serbia	40	0	1.0	0.0	10.0	10	1.0	1.0	0.1	10.0
Myanmar	40	0	0.0	0.0	1.0	10	0.0	0.0	0.0	10.0
Azerbaijan	30	0	0.0	0.0	0.0	10	0.0	10.0	0.0	10.0
Mexico	30	0	1.0	0.0	1.0	10	0.0	1.0	0.0	0.0
Nepal	30	0	0.0	0.0	0.0	10	0.0	10.0	0.0	0.0
Cyprus	30	0	0.0	0.0	0.0	10	0.0	1.0	0.1	10.0
Haiti	30	0	1.0	0.0	0.0	10	0.0	1.0	0.1	10.0
Tanzania	20	0	0.0	0.0	1.0	10	0.0	0.1	0.0	0.0
Saint Helena	20	0	1.0	0.0	10.0	10	1.0	10.0	0.0	0.0
Somalia	20	0	1.0	1.0	1.0	10	1.0	1.0	0.1	0.0
Jordan	20	0	1.0	0.0	1.0	10	0.0	1.0	0.1	0.0
China	20	0	1.0	0.0	0.0	10	0.0	0.0	0.1	10.0
Philippines	20	0	0.0	0.0	1.0	10	0.0	0.1	0.0	10.0
Kazakhstan	20	0	0.0	0.0	1.0	10	0.0	1.0	0.0	10.0
Antigua and Barbuda	20	0	0.0	0.0	0.0	10	1.0	10.0	0.0	0.0
Maldives	20	0	0.0	0.0	10.0	0	0.0	10.0	0.0	0.0
Laos	20	0	0.0	0.0	0.0	10	0.0	1.0	0.0	10.0
Japan	20	0	1.0	0.0	1.0	10	0.0	0.0	0.0	0.0
Angola	20	0	1.0	0.0	0.0	10	0.0	0.0	0.0	0.0
Chad	20	0	0.0	0.0	1.0	10	0.0	1.0	0.0	0.0
New Zealand	20	0	0.0	0.0	1.0	10	0.0	1.0	0.0	10.0
Mongolia	20	0	0.0	0.0	0.0	10	0.0	1.0	0.0	0.0
Fiji	20	0	0.0	0.0	0.0	10	0.0	1.0	0.0	0.0
Jamaica	20	0	0.0	0.0	1.0	10	0.0	1.0	0.0	10.0
St. Vincent and the Grenadines	20	0	0.0	0.0	0.0	0	0.0	10.0	0.0	10.0
St. Lucia	20	0	0.0	0.0	0.0	10	0.0	10.0	0.0	0.0
Georgia	20	0	0.0	0.0	1.0	0	0.0	1.0	0.0	0.0
Armenia	20	0	0.0	0.0	0.0	10	1.0	10.0	0.0	10.0
Cyprus	20	0	1.0	0.0	0.0	0	0.0	0.0	0.0	0.0
Belize	20	0	1.0	0.0	0.0	10	1.0	1.0	0.0	10.0
Zimbabwe	20	0	1.0	0.0	0.0	10	0.0	0.1	10.0	10.0

Figure 1: First Dataset Part 1

Artificial intelligence - corona\_latest.csv

number	Country/State	totalCases	newCases	totalDeaths	newDeaths	totalRecovered	newRecovered	ActiveCases	serious	critical	test	casesPer100000	deathsPer100000
1	USA	1688111	12212.0	1040.0	3964.0	12796.0	14917.0	4171.0	141112.0	1	2000.0	0.0	
2	Spain	168811.0	17739.0	0.0	16791.0	8770.0	7071.0	7668.0	156.0	18888.0	0.0	12810.0	
3	Italy	126553.0	13899.0	0.0	14211.0	18220.0	1461.0	1288.0	127.0	116131.0	0.0	14788.0	
4	France	112581.0	11439.0	0.0	27766.0	17622.0	4642.0	18618.0	121.0	111887.0	0.0	11218.0	
5	Germany	117766.0	2672.0	0.0	16768.0	14852.0	498.0	1215.0	16.0	117701.0	0.0	12788.0	
6	UK	84279.0	18612.0	0.0	0.0	17022.0	1289.0	1241.0	116.0	112974.0	0.0	1288.0	
7	China	82148.188	1741.0	2.0	17943.0	1250.0	121.0	17.0	2.0	0.0	0.0	0.0	
8	Japan	71280.0	4715.0	0.0	14266.0	12722.0	2078.0	401.0	12.0	142338.0	0.0	12128.0	
9	Sweden	14976.0	1139.0	0.0	1446.0	12222.0	186.0	470.0	12.0	171018.0	0.0	1409.0	
10	Belgium	27647.0	7689.0	0.0	1442.0	17554.0	1221.0	1258.0	121.0	142111.0	0.0	1814.0	
11	Netherlands	25281.0	17717.0	0.0	2828.0	12689.0	1286.0	14418.0	116.0	141524.0	0.0	14918.0	
12	South Korea	23412.0	1181.0	0.0	12788.0	11887.0	286.0	1917.0	119.0	117088.0	0.0	12110.0	
13	Canada	21281.0	717.0	0.0	7172.0	13474.0	587.0	144.0	17.0	121288.0	0.0	11188.0	
14	Brazil	12111.131	1128.0	7.0	171.0	12811.0	176.0	181.0	0.0	14188.0	0.0	176.0	
15	Portugal	11881.0	768.0	0.0	2711.0	11888.0	288.0	1407.0	10.0	14111.0	0.0	14888.0	
16	Australia	11779.0	118.0	0.0	1211.0	14249.0	0.0	181.0	0.0	118888.0	0.0	1211.0	
17	India	11765.0	768.0	0.0	1487.0	1488.0	143.0	1448.0	18.0	141877.0	0.0	14888.0	
18	Israel	11111.0	181.0	0.0	1411.0	1411.0	1411.0	1411.0	1411.0	14111.0	0.0	14111.0	
19	S. Korea	10817.125	117.0	1.0	7647.0	1287.0	11.0	186.0	0.0	141411.0	0.0	14888.0	
20	Canada	10481.0	181.0	0.0	161.0	1481.0	148.0	1481.0	18.0	14788.0	0.0	1418.0	
21	Latvia	9481.0	154.0	0.0	15.0	1476.0	174.0	1751.0	18.0	14888.0	0.0	14758.0	
22	Spain	8481.125	11.0	0.0	1481.0	1481.0	0.0	1.0	0.0	14111.0	0.0	147.0	
23	Peru	7919.0	191.0	0.0	1788.0	1481.0	114.0	128.0	0.0	14888.0	0.0	1218.0	
24	Colombia	7666.0	111.0	0.0	1481.0	1481.0	148.0	1411.0	11.0	14111.0	0.0	1118.0	
25	Japan	7128.0	11.0	0.0	1764.0	1481.0	128.0	148.0	1.0	14111.0	0.0	1411.0	
26	China	7111.0	18.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
27	Poland	6471.0	11.0	0.0	1471.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
28	Germany	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
29	Australia	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
30	Germany	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
31	China	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
32	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
33	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
34	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
35	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
36	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
37	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
38	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
39	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
40	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
41	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
42	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
43	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
44	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
45	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
46	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
47	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
48	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
49	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
50	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
51	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
52	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
53	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
54	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
55	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
56	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
57	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
58	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
59	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
60	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
61	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
62	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
63	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
64	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
65	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
66	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
67	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
68	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
69	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
70	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
71	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
72	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
73	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
74	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
75	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
76	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
77	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
78	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
79	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
80	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
81	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
82	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
83	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
84	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
85	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
86	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
87	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
88	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
89	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
90	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
91	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
92	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
93	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
94	Poland	6111.0	11.0	0.0	1481.0	1481.0	148.0	148.0	1.0	14111.0	0.0	1411.0	
95	Poland												

which is a csv file and then we gave a label to each column which is a number in our data set then we tested it and got the predicted data of how many people in each country is sick with COVID-19 and the actual data which is in the data set and also we calculated the distances between them . Here the K we used is 9 to get the nearest 9 neighbors.

This was the code used using KNN algorithm .

```

from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing
import pandas as pd

data = pd.read_csv('corona_data.csv', delimiter=',', header=0, columns=['name', 'Country', 'Time', 'TotalCases', 'NewCases', 'TotalDeaths', 'NewDeaths', 'TotalRecovered', 'NewRecovered', 'Serious', 'Critical', 'Fat Cases/24 hrs', 'Deaths/24 hrs'])

# name is the label
label = preprocessing.LabelEncoder()

# name is the label
name = label.fit_transform(list(data['name']))
country = label.fit_transform(list(data['Country']))
time = label.fit_transform(list(data['Time']))
totalC = label.fit_transform(list(data['TotalCases']))
newC = label.fit_transform(list(data['NewCases']))
totalD = label.fit_transform(list(data['TotalDeaths']))
newD = label.fit_transform(list(data['NewDeaths']))
totalR = label.fit_transform(list(data['TotalRecovered']))
newR = label.fit_transform(list(data['NewRecovered']))
serious = label.fit_transform(list(data['Serious']))
critical = label.fit_transform(list(data['Critical']))
fatcases = label.fit_transform(list(data['Fat Cases/24 hrs']))
deaths = label.fit_transform(list(data['Deaths/24 hrs']))

predict = "Class"

# 9 for training k = 9 for labels
X = list(zip(name, country, time, totalC, newC, totalD, newD, totalR, newR, serious, critical, fatcases, deaths))
y = list(label)

# Training & Testing data are as used & it is to separate the data
x_train, x_test, y_train, y_test = sklearn.model_selection.train_test_split(X, y, test_size=0.3)

# model training data
model = KNeighborsClassifier(n_neighbors=9)
  
```

Figure 4: KNN algorithm code Part 1



```
Artificial Intelligence - KNN.py

# Importing the dataset
dataset = load_csv('dataset.csv')

# Splitting the dataset into the training set and the test set
train_data, test_data = train_test_split(dataset, test_size=0.2, random_state=0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
train_data = sc.fit_transform(train_data)
test_data = sc.transform(test_data)

# Creating a K-Nearest Neighbors classifier
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5)

# Training the K-Nearest Neighbors classifier on the training set
knn.fit(train_data, train_data['label'])

# Making predictions on the test set
test_labels = knn.predict(test_data)

# Evaluating the model
accuracy = knn.score(test_data, test_labels)
print('Accuracy: %f' % accuracy)
```

Figure 5: KNN algorithm code Part 2

```
Artificial Intelligence - KNN.py

# Importing the dataset
dataset = load_csv('dataset.csv')

# Splitting the dataset into the training set and the test set
train_data, test_data = train_test_split(dataset, test_size=0.2, random_state=0)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
train_data = sc.fit_transform(train_data)
test_data = sc.transform(test_data)

# Creating a K-Nearest Neighbors classifier
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5)

# Training the K-Nearest Neighbors classifier on the training set
knn.fit(train_data, train_data['label'])

# Making predictions on the test set
test_labels = knn.predict(test_data)

# Evaluating the model
accuracy = knn.score(test_data, test_labels)
print('Accuracy: %f' % accuracy)
```

Figure 6: KNN algorithm code Part 3

#### 4.1.2 Decision tree using First Dataset

Decision Tree : this algorithm depends on dividing the dataset into sub classes and keep doing this method till it reach the classifier it is one of the supervised algorithms. Here we used it to divide each country with it's own data from deaths, cases and critical cases.

## 4.2 Second Dataset

The second dataset used consists of 210 country and their continents. Each country has its own data which are population, total cases, total death, new death, total recovered, new recovered, active cases, serious/critical cases, total cases/1M pop, deaths/1M pop, total tests, tests/1M pop.

Country/Region	Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeaths	TotalRecovered	NewRecovered	ActiveCases	Serious,Critical	Tot Cases/1M pop	Deaths/1M pop	TotalTests	Tests/1M pop
USA	North Am	3.31E+08	2007449		112469		761708		1133272	16923	6067	340	21251677	64349
Brazil	South Am	2.12E+08	691962		37312	813	302084		352566	8318	3257	176	999836	4706
Russia	Europe	1.46E+08	467973		5859		226731		235083	2300	3205	40	12721549	87175
Spain	Europe	46753690	288620		27136				617	6173	580	4063443	86920	
UK	Europe	67863668	286134		40542				604	4217	597	5581073	82239	
India	Asia	1.38E+09	258090	604	7207		124095	247	126788	8944	187	5	4774434	3462
Italy	Europe	60467085	234986		33899		165837		35262	287	3886	561	4236535	70063
Peru	South Am	32941004	196515		5485		86219		104811	1062	5966	166	1191956	36185
Germany	Europe	83787456	165669		8776		169000	500	7493	568	2219	305	4346880	51936
Iran	Asia	83921387	171789		8281		134349		29159	2596	2047	99	1084857	12927
Turkey	Asia	84279639	170132		4692		137969		27471	613	2019	56	2338593	27748
France	Europe	65264696	153977		29155		70842		53980	1053	2359	447	1384633	21216
Chile	South Am	19105644	134150		2190		95631		36929	1558	7021	115	708773	37098
Mexico	North Am	1.29E+08	171705	3484	13699	188	84413	2875	18985	378	949	106	536395	2611
Pakistan	Asia	2.21E+08	105671		4728	65	34355		67249	111	470	9	705632	3200
Saudi Arabia	Asia	34776977	101914		712		72817		26385	1564	2931	20	958237	27554
Canada	North Am	37720832	95699		7800		54233		33666	1816	2537	207	1896822	50286
Qatar	Asia	2807805	68790		54		44338		24398	245	24500	19	255533	91008
Bangladesh	Asia	1.65E+08	65769		888		13903		50978	1	400	5	397987	2418
Belgium	Europe	11556489	59526		8595		10340		16291	512	828	91670	80755	
Belarus	Europe	9449509	48630		269		23647		24714	92	5146	28	622313	65857
South Africa	Africa	59258991	48285		998		24364		22923	208	815	17	920064	15526
Netherlands	Europe	17132558	47574		6013					97	2777	351	388113	12654
Sweden	Europe	10095291	44730		4659				198	4431	462	275500	27290	
Ecuador	South Am	17624879	43120		3921		21020		18479	222	3447	205	117414	7229
Colombia	South Am	50847406	59326		1259		15522		22655	325	772	25	410719	8077
UAE	Asia	9882493	38808		276		21806		16726	1	3927	28	2500000	252973
Singapore	Asia	5847399	37910		25		24886		12999	3	6483	4	408495	69859
Portugal	Europe	10188446	34693		1479		20995		12219	58	3402	145	873998	85699
Egypt	Africa	101764071	34079		1237		8961		23881	41	333	12	159000	1321
Kuwait	Asia	4266300	31048		264		20205		11379	196	3465	62	515385	73961
Indonesia	Asia	2.73E+08	31186		3851		10498		18837		114	7	409992	1485
Switzerland	Europe	8650592	30965		1921		28700		344	28	3580	222	423468	48952
Ukraine	Europe	43748704	26999		788		12054		14157	304	617	18	424046	9693
Poland	Europe	37849071	26561		1157		12855		12549	160	702	31	1056396	27911
Ireland	Europe	459162	25201		1679		22898		824	36	5107	340	548616	70613
Argentina	South Am	45348962	22794		664		6909		15221	274	505	15	139323	4289
Philippines	Asia	1.09E+08	21895		1003		4530		16362	82	200	9	430081	3928
Afghanistan	Asia	38865373	20917	575	369	12	2171	296	18377	19	538	9	48305	1243

Figure 7: Second Dataset Part 1

Romania	Europe	19244947	20479		1333		14638		4508	144	1064	69	503200	26147
Dominican Republic	North Am	10840829	19600		538		12007		7055	110	1808	50	94511	8718
Israel	Asia	9197590	17915	52	298		15102	11	2515	27	1948	32	662916	72075
Japan	Asia	1.26E+08	17141		919		15139		1088	98	138	7	314483	2486
Austria	Europe	9003169	16862		672		15765		437	17	1877	75	409597	54381
Oman	Asia	5097088	16882		75		3451		13356	75	3312	15	108679	21322
Panama	North Am	4310139	16425		393		10218		5814	84	3811	91	72697	16887
Bahrain	Asia	1696917	14763		26		9468		5269	13	8700	15	367056	216308
Bolivia	South Am	11662351	13643	285	465	11	2086	184	11092	3	1170	40	36530	3132
Armenia	Asia	2962906	13130		200		4014		8916	10	4431	68	89171	23346
Kazakhstan	Asia	18761857	12399	165	56	3	7376		5427	42	685	3	945089	51439
Nigeria	Africa	2.06E+08	12486		354		3959		8173	7	61	2	76802	373
Iraq	Asia	40157701	12366		346		5186		6834	69	308	9	303053	7547
Denmark	Europe	5790943	11948		589		10755		604	15	2063	102	709442	121991
Serbia	Europe	8759407	11823		249		11348		226	15	1353	28	272502	31181
S. Korea	Asia	51266523	11814	38	273		10563	11	978	15	230	5	1018214	19861
Algeria	Africa	43796100	10154		707		6717		2730	24	232	16		
Moldova	Europe	4034514	9700		341		5638		3721	335	2404	85	63328	15697
Ghana	Africa	31026938	9638		44		3636		5958	3	311	1	233734	7533
Czechia	Europe	10707772	9628		327		6891		2410	12	899	31	471521	44035
Norway	Europe	5418531	8547		238		8138		171	5	1577	44	257303	47486
Malaysia	Asia	32183959	8322		117		6674		1551	5	257	4	610028	18864
Morocco	Africa	36881658	8224		208		7364		652	15	223	6	305953	8296
Cameroon	Asia	26497531	7908		212		4735		2961	28	298	8		
Azerbaijan	Asia	10133277	7553		88		4149		3316	66	745	9	128385	32407
Australia	Australia	25480378	7265	5	102		6706	3	457	3	285	4	1632116	64054
Guatemala	North Am	17852493	7055	263	252	22	1261	128	5542	5	394	14	31427	1756
Finland	Europe	5540198	6981		323		5800		658	6	1240	58	201000	36280
Honduras	North Am	9893859	6327	172	258	8	712	15	5357	13	639	26	21540	2177
Sudan	Africa	43775253	6081		359		2014		3708		139	8	401	9
Tajikistan	Asia	9522265	4529		48		2673		1808		476	5		
Uzbekistan	Asia	33486477	4552	21	17		3554		961	7	130	0.5	748555	22387
Senegal	Africa	16711254	4328		49		2588		1691	15	259	3	49180	2943
Djibouti	Africa	987030	4207		28		1877		2302		4262	28	35185	35627
Guinea	Africa	13106296	4117		23		2877		1217	24	314	2	14407	1099
Luxembourg	Europe	625281	4039		110		3899		30	1	6459	176	88051	140818
DRC	Africa	89353804	4016		85		537		3394		45	1		
Hungary	Europe	9661789	4008		546		2279		1183	21	415	57	210202	21796
Ivory Coast	Africa	26326544	3739		36		1818		1485		142	1	31549	1196
Nepal	Asia	29100465	3448		13		467		2968		118	0.4	241254	8290

Figure 8: Second Dataset Part 2

Romania	Europe	19244947	2047		1333		14638		4508	144	1064	69	503200	26147	
Dominican Republic	North Am	1084829	1960		538		12007		7055	110	1808	50	94511	8718	
Israel	Asia	9397590	17915	52	298		15102		11	2515	27	1948	32	662916	72075
Japan	Asia	1.26e+08	17141		916		15139			1086	98	136	7	314483	2486
Austria	Europe	9003169	16902		672		15793			437	17	1877	75	489597	54381
Oman	Asia	5097088	16882		75		3451		13356	75	3312	15	108879	21322	
Panama	North Am	4310139	16425		393		10218		5814	84	3811	91	72997	16887	
Bahrain	Asia	1696917	14763		26		9468		5309	13	8700	15	387056	23638	
Bolivia	South Am	11662351	13643	285	465	11	2086		184	11092	3	1170	40	36530	3132
Armenia	Asia	2962906	13130		200		4014			8916	10	4431	68	69171	23346
Kazakhstan	Asia	18761857	12859	165	56	3	7376			5427	62	685	3	965089	51439
Nigeria	Africa	2.06e+08	12486		354		3959			8173	7	61	2	76802	373
Iran	Asia	40157701	12366		346		5186			6834	69	308	9	302053	7547
Denmark	Europe	5790543	11948		589		10755			604	15	2063	102	709442	121991
Serbia	Europe	8739407	11823		249		11348			226	15	1353	28	272502	31181
S. Korea	Asia	51266523	11814	38	273		10563		11	978	15	230	5	1018214	19861
Algeria	Africa	43796100	10154		707		6717			2730	24	232	16		
Moldova	Europe	4034514	9700		341		5638			3721	335	2404	85	63328	15697
Ghana	Africa	31026938	9638		44		3636			5958	3	311	1	233734	7533
Czechia	Europe	10707772	9628		327		6891			2410	12	899	31	471521	44035
Norway	Europe	5418531	8547		238		8138			171	5	1577	44	257303	47486
Malaysia	Asia	32138393	8322		117		6674			1531	5	257	4	610038	18864
Morocco	Africa	36881658	8224		208		7364			652	15	223	6	305953	8296
Cameroon	Africa	26497531	7908		212		4735			2961	28	298	8		
Azerbaijan	Asia	10233277	7553		88		4149			3316	66	745	9	303885	32407
Australia	Australia	25480378	7265	5	102		6706		3	457	3	285	4	1632116	64054
Guatemala	North Am	17852493	7055	263	252	22	1261		128	5542	5	394	14	31427	1756
Finland	Europe	5540158	6981		323		5800			858	6	1260	58	201000	36280
Honduras	North Am	9893859	6327	172	258	8	712		15	5357	13	639	26	21540	2177
Sudan	Africa	43775253	6081		359		2014			3708		139	8	401	9
Tajikistan	Asia	9522265	4529		48		2673			1908		476	5		
Uzbekistan	Asia	33464677	4352	21	17		3354			981	7	130	0.5	748555	22387
Senegal	Africa	16711294	4328		49		2588			1691	15	259	3	49180	2943
Djibouti	Africa	987030	4207		28		1877			2302		4262	28	35165	35627
Guinea	Africa	13126296	4117		23		2877			1217	24	314	2	14407	3099
Luxembourg	Europe	625281	4039		110		3899			30	1	6459	176	88051	140818
DRC	Africa	89533804	4016		85		537			3394		45	1		
Hungary	Europe	9661789	4008		546		2279			1183	21	415	57	210202	21756
Ivory Coast	Africa	26330544	3739		36		1818			1885		142	1	31549	1198
Nepal	Asia	29100465	3448		13		467			2968		118	0.4	241254	8290

Figure 9: Second Dataset Part 3

#### 4.2.1 K-means algorithm using second dataset

K-means : This algorithm is used for unsupervised learning in which it clusters the data by grouping the clusters comparable data. The K here is constant we choose it. Here we used it to group all the data for each country .

#### 4.2.2 Support vector machines algorithm using second dataset

SVM: stands for " Support vector machines " it separates 2 classes by a straight line (can be viewed the visualizer). a device called "kernel trick" used when the classes cannot be separated to make boundaries of different shaped.

## 5.1 KNN algorithm

The results of KNN algorithm appears as follows, as all countries are retrieved with their numbers only but not their names. Each country retrieve the data of its predicted number of infected patients and the real number of infected patients.

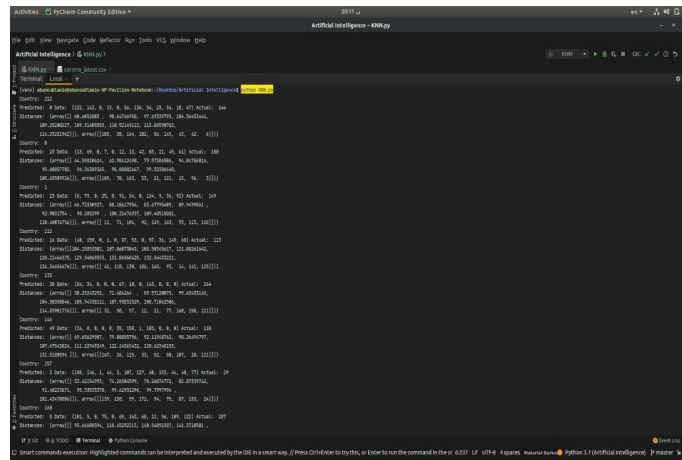


Figure 10: KNN Results Part 1

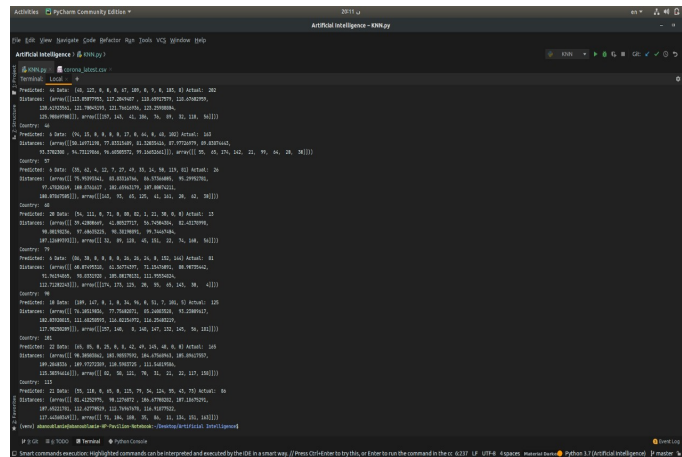


Figure 11: KNN Results Part 2

## 5.2 Decision tree algorithm

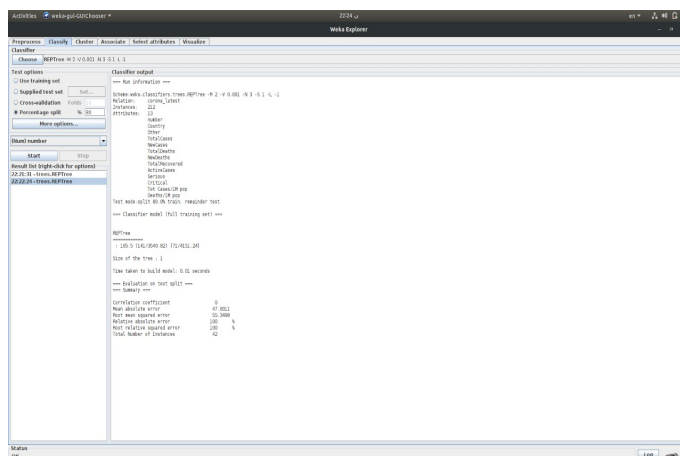


Figure 12: Decision tree algorithm results with split 80%

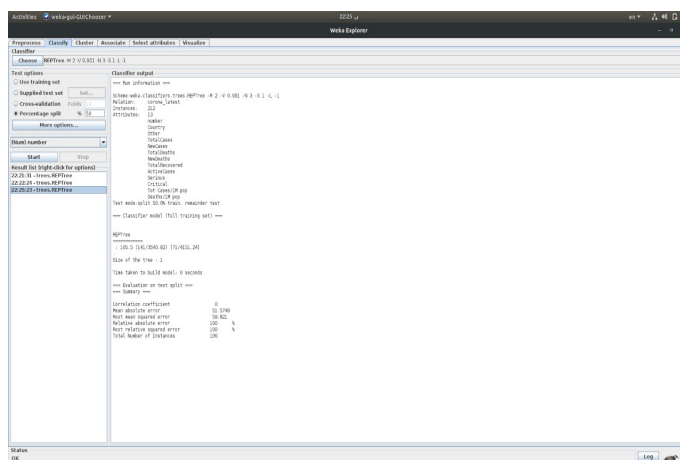


Figure 13: Decision tree algorithm results with split 50%

### 5.3 K-means Algorithm

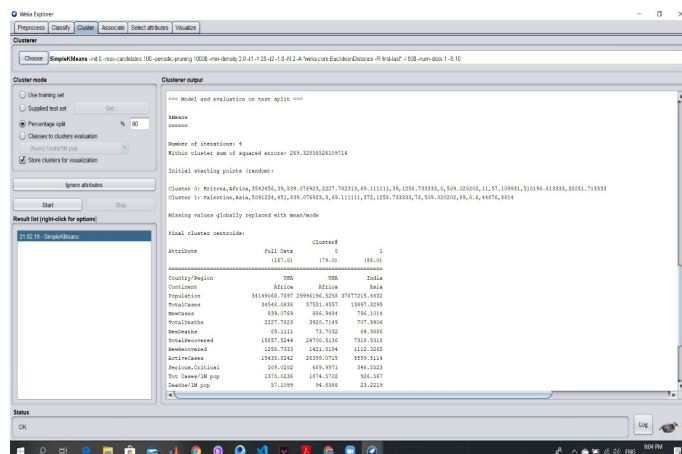


Figure 14: K-means results Part 1

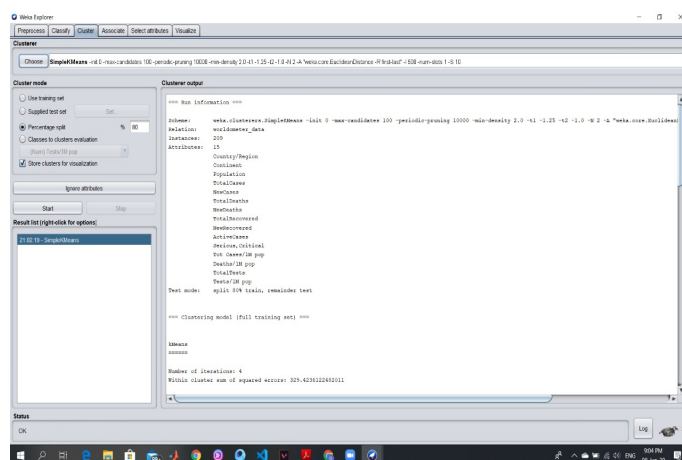


Figure 15: K-means results Part 2

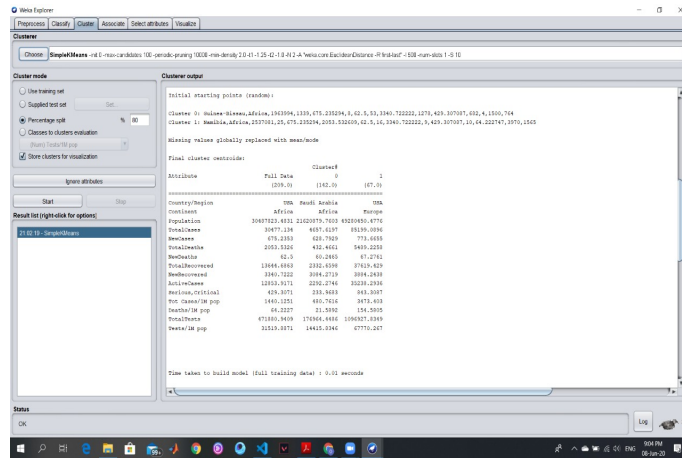


Figure 16: K-means results Part 3

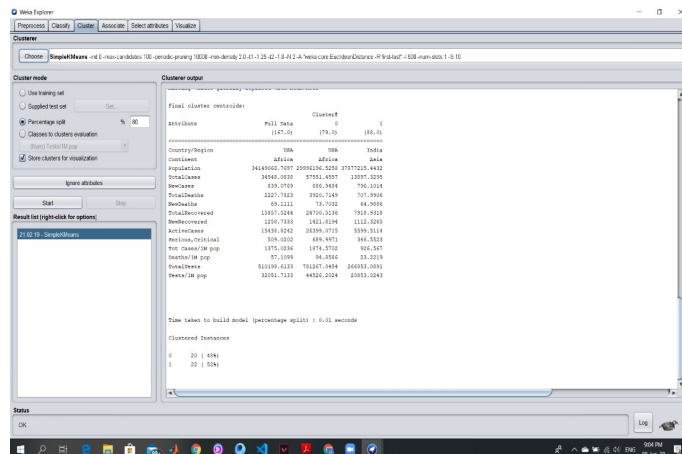


Figure 17: K-means results Part 4

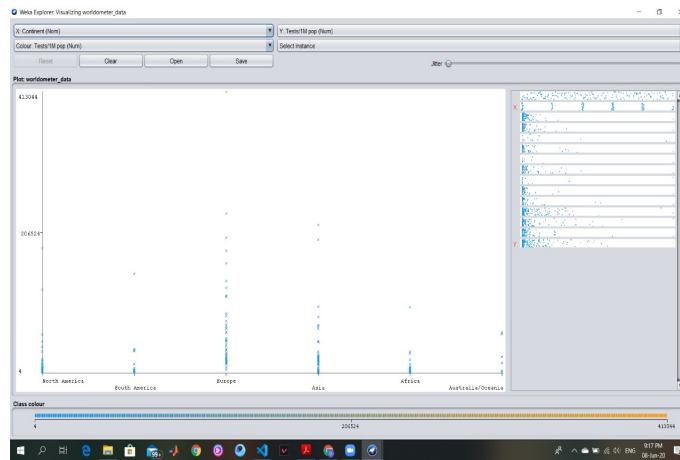


Figure 18: K-means results visualize

## 5.4 SVM algorithm

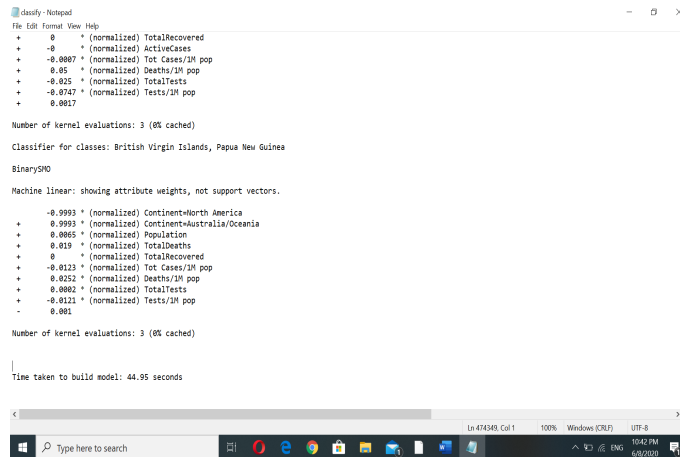


Figure 19: SVM results : Time taken to build model



```

classify - Notepad
File Edit Format View Help
+ -0.3598 * (normalized) TotalTests
+ -0.0475 * (normalized) Tests/3H pop
+ 1.0112

Number of kernel evaluations: 3 (0% cached)

Classifier for classes: USA, Russia
BinarySVM

Machine linear: showing attribute weights, not support vectors.
-0.3541 * (normalized) Continent=North America
+ 0.3541 * (normalized) Continent=Europe
+ -0.0475 * (normalized) Population
+ -0.2744 * (normalized) TotalCases
+ -0.3376 * (normalized) TotalDeaths
+ -0.2516 * (normalized) TotalRecovered
+ -0.2613 * (normalized) ActiveCases
+ -0.3865 * (normalized) Serious,Critical
+ -0.0459 * (normalized) Tot Cases/3H pop
+ -0.0835 * (normalized) Deaths/3H pop
+ -0.1418 * (normalized) TotalTests
+ 0.0173 * (normalized) Tests/3H pop
+ 0.9907

Number of kernel evaluations: 3 (0% cached)

Classifier for classes: USA, Spain
BinarySVM

Machine linear: showing attribute weights, not support vectors.

```

Figure 20: SVM results : Comparing between USA and Russia

```

classify - Notepad
File Edit Format View Help
+ -0.2394 * (normalized) TotalTests
+ -0.0396 * (normalized) Tests/3H pop
+ 0.7651

Number of kernel evaluations: 3 (0% cached)

Classifier for classes: USA, Ivory Coast
BinarySVM

Machine linear: showing attribute weights, not support vectors.
-0.2457 * (normalized) Continent=North America
+ 0.2457 * (normalized) Continent=Africa
+ -0.0443 * (normalized) Population
+ -0.2453 * (normalized) TotalCases
+ -0.2456 * (normalized) TotalDeaths
+ -0.2451 * (normalized) TotalRecovered
+ -0.2454 * (normalized) ActiveCases
+ -0.2396 * (normalized) Serious,Critical
+ -0.0838 * (normalized) Tot Cases/3H pop
+ -0.0647 * (normalized) Deaths/3H pop
+ -0.2453 * (normalized) TotalTests
+ -0.0328 * (normalized) Tests/3H pop
+ 0.7635

Number of kernel evaluations: 3 (0% cached)

Classifier for classes: USA, Greece
BinarySVM

Machine linear: showing attribute weights, not support vectors.

```

Figure 21: SVM results : Comparing between USA and Ivory Coast

## 6 Conclusion

This document was made to discuss the difference between the number of infected patients between all countries. We wanted to facilitate this process by using some algorithms which are KNN, SVM, K means and Decision tree. KNN algorithm was used by Python, while SVM, K-means and Decision tree algorithms was used with Weka.

## References

- [1] Mohamad Chahrour et al. “A bibliometric analysis of Covid-19 research activity: A call for increased output”. In: *Cureus* 12.3 (2020).
- [2] Yawei Chang and Houquan Liu. “Semi-supervised classification algorithm based on the KNN”. In: (2011), pp. 9–12.
- [3] Chenlin GU et al. “Mathematical recommendations to fight against COVID-19”. In: *Available at SSRN 3551006* (2020).
- [4] Nirmal Kandel et al. “Health security capacities in the context of COVID-19 outbreak: an analysis of International Health Regulations annual report data from 182 countries”. In: *The Lancet* (2020).
- [5] Mary A Lake. “What we know so far: COVID-19 current clinical knowledge and research”. In: *Clinical Medicine* 20.2 (2020), p. 124.
- [6] Canelle Poirier et al. “The Role of Environmental Factors on Transmission Rates of the COVID-19 Outbreak: An Initial Assessment in Two Spatial Scales.” In: *Available at SSRN 3552677* (2020).
- [7] Max Roser et al. “Coronavirus disease (COVID-19)–Statistics and research”. In: *Our World in data* (2020).
- [8] Fei Zhou et al. “Clinical course and risk factors for mortality of adult in-patients with COVID-19 in Wuhan, China: a retrospective cohort study”. In: *The lancet* (2020).

## 7 Other References

- 1. [https://www.saedsayad.com/k\\_nearest\\_neighbors.htm#:~:text=K%20nearest%20neighbors%20is%20a,as%20a%20non%2Dparametric%20technique](https://www.saedsayad.com/k_nearest_neighbors.htm#:~:text=K%20nearest%20neighbors%20is%20a,as%20a%20non%2Dparametric%20technique)
- 2. <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
- 3. <https://www.edureka.co/blog/k-means-clustering/>
- 4. <http://www.onmyphd.com/?p=k-means.clustering>
- 5. [https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200225-sitrep-36-covid-19.pdf?sfvrsn=2791b4e0\\_2](https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200225-sitrep-36-covid-19.pdf?sfvrsn=2791b4e0_2)
- 6. <https://www.kaggle.com/balaaje/coronavirus-covid19-dataset>
- 7. <https://www.kaggle.com/imdevskp/corona-virus-report>